AMENDMENTS TO THE WRITTEN DESCRIPTION

Please add the following new paragraph prior to the paragraph beginning on line 3 of page 1:

This application is a National Stage of International Application No. PCT/SG98/00102, filed on December 16, 1998.

Please replace the paragraph on page 1 at lines 6 - 13 with the following amended paragraph:

Recent years have seen a number of developments in computing science regarding how elements within a software application are treated and handled. In this context the most basic elements to be found within a software application are data and program modules. Traditional procedural programming paradigms focus on the logic of the software application, so a program is structured based on program modules. One uses the program modules by explicitly supplying to them the data on which they should operate. A potential pitfall of this paradigm is that it is difficult to guarantee type safety, ~~ie~~ i.e., that the data being passed to a program module is of the correct type.

Please replace the paragraph bridging pages 1 and 2 with the following amended paragraph:

While the object-oriented paradigm represents a significant advance in software engineering, the data and modules that constitute each object are static. The paradigm is still inadequate for writing programs that must evolve during execution, ~~eg~~ e.g., programs that need to pick up, drop, or substitute selected modules. There have been several attempts at overcoming this limitation. For example, work described in US Patents 4954941, 5175828, 5339430 and 5659751 address techniques for re-linking or re-binding selected software modules dynamically during runtime. Also Microsoft's Win32 provides for explicit mapping and un-mapping of dynamic linked libraries into the address space of a process through the LoadLibrary and FreeLibrary calls. With this prior art, however, the prototype or specification of functions and symbols are fixed beforehand and compiled into application programs. This means that an object cannot invoke a module of another object for which the specification is not known at compile time.